Homework 03

Due: 11:59 pm, Thursday March 7

Please handin your homework assignment on gradescope as a PDF file with each problem on a separate page.

Web Security

Problem 1

A common way that web application developers check that user inputs are valid (for example, that they adhere to password length requirements, that emails are properly formatted, etc.) is with javascript executing in the browser. When a form is submitted, the javascript first checks to make sure that the inputs are valid, and refuses to submit the form if they are not. Explain why a web site cannot assume that the input it receives from users will be valid as a result of these checks.

Problem 2

For this problem you may want to look up details online, but make sure to cite your sources. See the collaboration policy for details.

In this hands-on problem, you'll explore some of the details of HTTP. In particular, you're going to navigate to https://cs.brown.edu on a browser and analyze HTTP request and response headers by using the developer tools of your browser (e.g., the Chrome DevTools) and going to the "Network" tab.

a) On a private browser window, navigate to https://cs.brown.edu.

   i. Write down the HTTP request headers for the main resource (/) and the headers for the corresponding HTTP response. If you are using LaTeX, consider using the `verbatim` environment.

   ii. Explain the meaning of each line of the above request and explain the meaning of as many as possible response headers.

b) Now sign in to https://cs.brown.edu and then navigate again to https://cs.brown.edu. Explain how the request and response differ from the case where you were not logged in (i.e, the case of question a). You do not need to submit the request nor response for this question.

Passwords

Problem 3

A website requires users to provide a password to access a secure area of the site. If a user forgets their password, they can reset it by correctly answering a security question (e.g., "What is your city of birth?") whose answer is selected by the user at the time of account setup.

a) Is this more or less secure than a website that only requires the password (and does not allow resetting via a security question)? Why?

b) Is this more or less secure than a website that requires both a password and a security question to log in every time (and does not allow resetting via a security question) ? Why?

c) Compare the following two approaches to selecting the answers to security questions in terms of security and usability.

- *Untruthful answers:* potentially valid answers that are not accurate, such as entering New York as your city of birth when you were instead born in San Francisco.
- *Random answers:* randomly generated strings of characters, such as entering `4%oS8–RXh;=` as your city of birth.

Problem 4

Gru needs your help designing a dating app for the minions. To gain access to the app, a minion would need to provide some sort of authentication. You have come up with two possible authentication systems:

a) Minions provide one password of length 9 (salted and hashed)

b) Minions provide two different passwords, each of length 8 (salted and hashed using two different salts). Minions have to enter the first password correctly and then enter the second password.

Assuming passwords are alphanumeric (upper case and lower case) and chosen uniformly at random, which system is more secure against an attacker who is capable of hashing 100 million passwords per second? Justify your answer analytically.

Problem 5

In practice, when people choose 10-character alphanumeric passwords, they tend to provide much weaker security than when computers generate 60-bit cryptographic keys. There are 62 possible characters (26 lower-case letters, 26 upper-case letters, and 10 digits) in an alphanumeric password, so there are 62^{10} possible 10-character alphanumeric passwords. There are 2^{60} possible 60-bit cryptographic keys. 62^{10} is approximately equal to 2^{60}, so you might expect them to be equally secure. Why, in practice, is this not the case?